MICROPOLIS/VECTOR GRAPHIC USERS GROUP

MUG Newsletter #51 - October 1984

Published Monthly by DAMAN
604 Springwood Circle, Huntsville AL 35803
Subscription rates:
North America; $18/year
Elsewhere; $25/year - Airmailed

For Assistance in the following areas, call:
(205) 881-1697 during the Central Times listed.
Micropolis Basic, general information:
- Lynn from 9-5.
MUG/Vector Graphic & whatever:
- Buzz from 7-10.
DAMAN Software & Hardware sales:
- Lynn from 9-5.

DISK EMULATORS

by Edward U. King
Casey & King, Attorneys at Law
PO Box 787, Ithaca NY 14850

I have been using an "Interstellar Drive" (ISD) by Pion, Inc. of 101R Walnut Drive, Watertown, Mass. 02172, with my Vector Graphic 2600, dual 5 1/4 inch Tanden "quad density" drives (100 TPI at a net usable 596K each formatted; but the B drive can be configured as a single-sided 300K drive emulating a Micropolis Mod II) CP/M 2.22H machine for about a year, and I am completely sold on it!

The incredible increase in speed of operation in reading from and writing back to the Pion Drive is a wonder to behold! Just bringing up MEMORITE, or the EPL EXECUPLAN spread sheet, or VERDICT (my professional time and billing program from MicroCraft) from the Pion Drive is a pleasure. Accessing the OVL (overlay) program modules of the time and billing system proceeds at ten times the speed of accessing the same from a floppy disk.

Using the "Get" command within ExecuPlan to extract information from other spread sheet files which have been pipped to the Pion Drive runs from, I would judge, ten to fifty times faster than the crunching which takes place in looking for that information and retrieving it from a floppy disk.

DESCRIPTION OF PION DRIVE

The Pion Drive comes in many configurations, for attachment to many different brands of computers -- including Apple, IBM, Vector Graphic, etc., etc. In my configuration for the Vector Graphic, there is an S100 card to be inserted in the computer, with a flat ribbon attachment therefrom to the external small metal box (8" wide x 9 1/2" deep x 3 1/2" high) which contains its own separate AC power cord-with-transformer (to produce 9.2 VAC, 2.5 Amp Output) with three small LEDs protruding from the front panel to indicate Power on, Read/Write activity. Inside the box is a set of NiCad Batteries to sustain power to the chips for twenty minutes in the event of an AC power failure. And, of course, there is the main Memory Board (256 K) plus the Optional Extender Board to permit the addition of up to three more 256K Memory Boards. (I started out with 512K and am now expanding to the full 1 Megabyte.)

The I/O is accomplished by using four contiguous parallel "ports" --- not of the RS-232 hardware variety, but of the mysterious, almost philosophical, Memory-mapped(?) variety which I read about but do not understand! Frequently I read that an 8-bit machine can have up to 256 Ports; and the Public Domain Program SURVEY3.COM indicates that I have active at any particular moment a varying number of active ports -- somewhere in the neighborhood of 127 to 163 --- or some other foolish number around that range.

My Vector Graphic hardware manual indicates that only one bank of 32 ports may be utilized at a time out of the 8 possible banks; and I deduced that, because my RS-232 accesses printer and modem with port 4 for data and 5 for status (that goes down 6 and 7, the manual says) that I must be using Bank I as described in the Hardware Manual. (I am told that I could cut and solder jumpers to select any of the other seven banks in place of this one --- but I have no earthly notion of why I should or might want to, since I can use only one at a time anyway! I am not sure I will ever really understand the subject of quartz! I keep reading "over my head" about these thing, and once in a while a bit of intelligible information seems to flake off on me, like dandruff; but it is impermanent and so far seems to be about as useless as dandruff!)

OTHER SOLID-STATE DRIVES

There are other solid-state drives, such as "Semi-Disk" and Compu-Pro's "M-Drive" -- which is described as a "warp drive" -- whatever that is. A favorable review of Semi-Disk by Prof. Alan R. Miller in IFAge Nov. 1982 warned that the Semi-Disk Install Program altered entry points for both BIOS and BDOS, which could cause problems with some software, making such utilities as Microshell, Badlim, etc. unusable. But Bill Machrone, who did an excellent and favorable review of Semi-Disk in the January issue of MicroSystems, points out that this can be avoided by utilizing the source code (which comes with the Semi-Disk software drivers) to achieve a relocation of the BIOS patches above CP/M, rather than integrating them into the BIOS.

The Pion Drive does not have this problem, since the 224K Drivers for it are either integrated into the top of your CBIOS (if you wish to follow the clear instructions given in the excellent Pion Drive manuals -- which tells you how to either make 1K available there above your printer drivers; or, if you have source code for CBIOS, to integrate the 224K Drivers into that); or you can do has I did -- - viz. locate the driver anywhere else in memory.

My 56K Vector Graphic has both my 4.2 Monitor ROM, the Memory-mapped Video area, and some other RAM residing above CP/M --- which occupies up to just below the Mon at E000. The upper most 1K of Memory (F000 - FFFF) apparently needs only its very last page (FF00 - FFFF) to contain some vital information, including a stack, the top of which is at FFD0. After much experimentation, I found I could live dangerously and locate a Pion Drivers at FC00. So far, I don't seem to have bumped into the System Stack. (I do know that the entire 1K, from F000 up, is utilized by the Monitor when the machine is first turned on -- it always has alot of garbage in there, even though I had killed all but the upper most reaches before shutting the machine off, just to see what would happen upon turn-on.)

In that article, Machrone also points out that the Semi-Disk choice of accessing memory through I/O ports rather via extended addressing is "just plain faster" than the Warp Drive extended-address system of selecting alternate pages of memory and flipping bytes back and forth through the processors registers; and he notes that the use of dynamic RAM chips makes Semi-Disk more compact and cheaper than the equivalent static chips necessary to a Warp Drive implementation.

The Semi-Disk reputedly has a capability of 8 Megabytes, while M-Drive can support 4 Megabytes! The Pion Drive is limited to 1 Megabyte.

I chose the Pion Inc. ISD over the Semi-Disk for 3 reasons: (1) It had built-in battery backup (giving you about 1/2 hour to get the current restored if power went off). (2) It had its own separate power

supply -- so you could leave material stored overnight with the rest of the compter shut down. (3) Its Driver code can be stuck anywhere in memory. Semi-Disk II has many improvements that make it rivalor perhaps surpass the ISD, but as far as I can tell the Driver still must reside below the CCP; and the battery backup costs $150 extra.

INSTALLATION

When I first received the Pion Drive and read the two Manuals (one for Hardware and the other for Software) I got the erroneous impression that I had to be able to write the Driver -- even though I am not a programer. I began to try to do it, using the information in the manual! But I had to telephone them, and was given the genial and welcome advise that I did not have to write anything: all I had to do was configure the two software programs (for FORMAT and ATTACH to show the capacity of the Pion Drive and the four consecutive ports selected by me -- selected via little metal jumpers supplied and easily moved to different locations on the Boards as easily as using a counter peg-board!)

This arrangement does save my wasting 3/4K of memory above CP/M (because I don't know how to integrate the driver into CBIOS without having source code); but it does mean that access to the drive must be arranged each time one turns the machine on. I used a cold-start-only-boot of the ATTACH program to make the drive available with relative ease. One advantage of using that separate driver and Attach program is that it provides an opportunity to Menu-select (Y/N) a quick CHECKSUM verification of the data on the ISD. (Remember, the separate power supply permits you to leave that drive powered up 24 hours a day, separately from the computer, so you can maintain basic utility and program files on the drive continously. You, however, are advised to disconnect the drive once per month to fully discharge the NiCad batteries in order to keep them active and make them fully rechargeable.

.....

Editor's Notes: Ed mentioned that PION stopped making the disk emulator about nine months ago because they could not get chips. Maybe they are back in business by now, maybe you'll see a used PION on the market. In any case, it's obvious that a disk emulator does install in a Vector, and this information will aid us in understanding how to select one for ourselves.

I don't know what the SURVEY3 program is, and I don't particularly understand all this "port" stuff either. However, I doubt that Ed is using 127 ports on his Vector. Judging from his article, he has four ports assigned to the PION, maybe 6 more for the RS-232 serial, 2 for the parallel, plus 2 on the Flashwriter board that handle the keyboard. I also believe that ports are ports, that is, there is no difference between a port that handles RS-232 serial and a port that handles parallel. What's connected to them is different.

You can have more than 32 ports installed. It's just that the Bitstreamer II can have its ports located at port locations that vary by 32. My documentation doesn't say 32, by the way. It says 16. (Base address of hexidecimal x2 to x9 where x can be any digit between 0 and 7)

You might want to do this if you had two different printers, a modem, a plotter, and a hardline to a separate computer all serially connected to your Vector. That's five devices. Since a Bitstreamer can only handle 3 devices, you'd need two Bitstreamers. Since they can't both be located at the same port addresses, you'd need to jumper one board to an optional base address.

Finally, the fact that the 1K area from F000 to FFFF shows data after zeroing it, shutting down, and turning on, doesn't mean the Vector is using it. Most memory boards (this system RAM is just a one-chip memory) will come up with random data when first turned on.

This was a terrifically interesting article to me. Many thanks to Ed for writing it.

.....

ONLY HERPES LASTS FOREVER
==========

by Burks A. Smith of Datasmith
PO Box 8036, Shawnee Mission KS 66208

I have noticed in recent months that the MUG is drifting away from MDOS, with more and more CP/M articles. I remember when some readers complained when the first of the CP/M articles appeared. I have also noted your editorial concern over the future of the MUG, recognizing that the number of MDOS and even Vector Graphic users are dwindling.

It seems at this point that a proper service for the MUG to perform is to help members make the transition from Micropolis equipment to the new generation of PC's. It seems likely that, at least for awhile, these members will probably keep their old equipment since no good market exists for obsolete computers. To this end, I have written a couple of columns outlining my thoughts for the future of personal computing and some methods to move data from the old machines to the new ones. By communicating with the new computers, the usefulness of the old boxes can be extended.

If there is interest among members for more of this type of thing, I can write about more advanced communications methods than the simple file transfers outlined. I'll wait and see if anyone cares.

THE DEATH OF CP/M AND THE Z-80

Just when you were worrying about converting from Micropolis MDOS to CP/M when your aging Micropolis-based system has to be replaced, IBM has changed the rules. By adopting Microsoft's MS-DOS operating system, IBM has effectively shut CP/M out of the new 16-bit desktop computer market. IBM and makers of IBM-compatibles have sold so many machines that a new standard has been adopted by default, just as CP/M was adopted as the standard for 8 bit machines.

It is only a matter of time until all the 8080 and Z-80 based machines are sent to the scrap heap, and CP/M will go with them. Nobody is going to build any more 8-bit computers, and most of the manufacturers of small computers have either already switched to 8086 and 8088 based machines or have gone out of business.

Improvements in microprocessor and random access memory technology have made the 8-bit 8080 or Z-80 chip in your computer obsolete. Back in 1978 when Micropolis MDOS version 3 was released, 64K of memory cost so much money that most micros got along with 32K. Later, memory got cheaper and 64K became standard equipment on most micros. Still, a Vector Graphic 64K dynamic RAM board cost a thousand dollars, a significant percentage of the cost of the whole system. Today, you can buy a complete, Z-80 based computer with 64K of memory that plugs in as an accessory for an IBM PC for $695, and dollars are worth less!

With lower memory cost, more sophisticated programs could be written without worrying much about how much memory they occupied, but the 8080 and Z-80 chips, like all 8-bit microprocessors, could only access 64K of memory directly. If you wanted more memory, you had to write programs that simulated a larger addressing scheme, which is very cumbersome and limited in power. The 8086 family of microprocessors can directly address up to 1 Megabyte of

memory. Most off-the-shelf machines today come standard with 128K of memory.

Digital Research Inc., CP/M's developer, is not going out of business. Rather than throw in the towel when it became apparent that CP/M-86 wasn't going to make it in the 16-bit world, DRI converted its line of excellent language products to run under PC-DOS and recently announced their own multi-tasking "Concurrent PC-DOS". DRI is still trying to get an operating system out that will compete with Microsoft's product, but it is apparent that it will at least be compatible with MS-DOS.

HELLO, MS-DOS

MS-DOS is Microsoft's name for the operating system now considered "standard" for all 8086 and 8088 microprocessor based computers, including the IBM PC. This operating system, originally started by Seattle Computer Corp. as a way to get their new 16-bit computer off the ground several years ago, was purchased by Microsoft and heavily influenced by IBM during development. Today, it is known by several names, inculding PC-DOS, but underneath it is the same product, licensed from Microsoft corp.

I remember reading somewhere that the original designer of MS-DOS hated CP/M and wanted to design a "better" operating system. However, IBM wanted to make the transition from CP/M to its new operating system as simple as possible so it mandated a high degree of compatibility with CP/M, including the familiar (to CP/M users) "A>" prompt, and very similar assembly language interface conventions. This was a good idea, since it allowed software developers to convert existing packages to the IBM PC with a minimum of time and expense, but it precluded any radical departures from the norm. As a result, MS-DOS version 1 looked very much like CP/M, at least on the surface. MS-DOS version 2 added many more features which CP/M did not have.

From an assembly language programmer's point of view, it is rather surprising to compare MS-DOS with Micropolis MDOS. The features that were added to MS-DOS version 2 to improve it were included in MDOS from the beginning. For example, CP/M made the user fill in a block of memory called a "File Control Block" (FCB) whenever a file was to be accessed. The FCB contained the file name, record number, extent number, and a lot of other stuff that was "reserved" for CP/M's use. Files were accessed as if they always had 128 byte records, and if you wanted to load a file into memory from a program, you had to write a program to read 128 byte chunks of the file and put them in the right place in memory. In addition, CP/M didn't have a facility that told you how long a file was. The information usually had to be found by trial and error.

Old MDOS and new MS-DOS features access file by file number and have a built-in routine that will pick a file name out of a command line instead of making you copy it byte-by-byte to an FCB. The FCB itself, although it exists somewhere, is not seen by the programmer. Files can be accessed as a stream of bytes or as records and an entire file can be loaded to or saved from a particular place in memory with a single command. The length of a file is known. MDOS still has MS-DOS beat in the area of many handy line parsing and utility commands, but MS-DOS has variable length records, and does all the housekeeping necessary to keep track of them regardless of the actual sector size on the disk. MS-DOS also has powerful memory management features that are related to the architecture of the 8086 microprocessor, allowing any program to load and execute another program and then return to itself.

Now that they have their very own operating system at the top of the heap, the Microsoft folks have improved Microsoft Basic. Called GW-Basic (for Gee Whiz) inhouse, it is also known as PC-Basic, Texas Instruments Professional Basic, Zbasic (by Zenith), and many others. GW-Basic is backward-compatible with Microsoft Basic-80 version 5 except in a few file-related functions. New features include a new family of music and graphics commands, but the most useful is the ability to open a device as a file. The printer is called "LPT1:" and the screen is called "SCRN:", much as "*P" and "*T" are used in Micropolis Basic. Formerly, Microsoft could only send output to a printer with an LPRINT statement. With GW-Basic, however, you can now also open the keyboard or a communications port.

MS-DOS is a very powerful operating system and is a pleasure to use when compared to CP/M. It is particularly well suited to the use of hard disks, which are getting so inexpensive that it is ridiculous not to have one if you are doing serious business computing. New releases of MS-DOS (version 3 is scheduled soon) promise to give multi-tasking and multi-user capability.

It is clear that MS-DOS will reign as the major microcomputer operating system for many years, probably much longer than CP/M was dominant. The 8-bit computers quickly were being used to their full memory and speed capacity in business applications. The 16 bit chips, with their megabyte memory capacity, are probably adequate for business use for the forseeable future. Unless a competing company like AT&T can knock IBM and its followers out of first place, MS-DOS or something very similar to it appears to be the safe bet, at least for the rest of the 1980's.

Next month: From MDOS and CP/M to MS-DOS, and how to do it.

.....

LANGUAGE FORUM
==========

by Jeffery Bell
252 Clark Hall, Cornell University, Ithica NY 14853

I have received eighteen replies to the Computer Language questionaire from the July newsletter. I was slightly disappointed at the small number of replies, but those of you who did reply have had interesting things to say -- enough to provide material for several short articles.

From the replies I did receive it seems to me that members of MUG can be sorted in just about equal numbers into two somewhat artificial catagories. In the first catagory are those of us who program infrequently and then usually in BASIC. These people are sometimes hobbyists but more often use computers in their business activities (lawyers, accountants, etc.) In the second catagory are those of us who do a moderate to large amount of work-related programming, as well as some hobby applications, using several different computer languages. Members of these two groups have very different experience and interests and I was surprised to see that we co-exist in one users' group.

Edward King is a good example of someone in the first catagory. He is a lawyer and fellow-MUGger from the same town as I live in (although we have never met). He says, "(I have) almost no language familiarity . . . Q: Why don't I program? A: Too time-consuming . . . Micro-Craft's Time and Billing program is a much better, more sophisticated program than I could achieve on my own in years! So too for a word-processor program: why should I try to write one?" Similarly, Orville Fudpucker says, "CBASIC plus Datastar, Wordstar, Supersort, Calcstar, Word+, etc. can do most all I want." Or as Marlin Weston put it: "I'm a user, not a programmer."

Related to these quotations is an interesting observation made by Dave DeDene. "Programming could become a dying art in a few years with the advent

of powerful fourth-generation tools. I work for a large corporation and have installed a number of these packages in the past year."

I am not very well informed about this area of computing and I would certainly like to hear more about this topic from Dave or other members. My only comment is that I can't imagine any database management-type programming tool taking the place of Fortran in the several scientific application that I am familiar with. Business is probably a different story.

Of those members who are in the second catagory, I was interested to find that several members have a broad knowledge of and a great deal of experience in computer languages. Those who have such knowledge could contribute a lot to MUG (hint, hint). They are probably more qualified to write about languages than I am -- but I realize that someone who writes computer programs for a living and plays with computers as a hobby may be too computer-saturated to also write about them.

Scott Duncan, definitely a programming-type member, raises an interesting issue. He asks, "Is there a language subgroup in MUG outside of BASIC (PL/I or C, perhaps)?"

No, there isn't. I am not very interested in PL/I (see below), however, I would love to learn more about C. Any others out there feel the same way?

I have given short-shift in my discussing to several computer languages because of my ignorance of them. Scott Duncan apparently thinks highly of PL/I. Because I think of it as a Pascal-like language I am not very interested. (Like to argue about that Scott?)

Dave DeDene and Scott Duncan mentioned COBOL. I have never heard anyone say anything good about COBOL.

In part six of the questionaire Dan Eisenreich asked, "What is your opinion of ADA?" and answered his own question: "Sounds like COBOL (yuk)." For those of you who do not know, ADA is the new "standard" language from the Department of Defense, the same people who are responsible for numerous technically-baroque weapons systems that work well, but only when treated with tender, loving care by highly-trained civilian technicians. I hear that ADA tries to be everything for everybody and is therefore rather complicated.

Several members have mentioned FORTH and I will try to say more about this interesting language in the future. Also, receiving some mention were Logo and LISP. Logo I am ignorant of. LISP is the darling of people who work in the fields of artificial intelligence and the solution of logical problems by computer. Again, I am far from my area of expertise.

Finally, Albert Folop wrote several enthusiastic comments about CMS-2. Unfortunately this is the first time I have heard about that computer language. If he or other members would care to fill me in about it I'd be interested.

To be continued.

.....

WHY SHOULD I HAVE A MANUAL?

by Lynn Rudow

Last year we started carrying the Micropolis disk parts, which threw me into a world almost as foreign as CP/M. But, being the sport I am, I figure if I can put up with CP/M, I can certainly learn the names of a few parts. Right? Besides, EVERYONE has that BIG manual that has the pictures of the drive parts and their name, that even has a separate list of the parts and their part number. Right?

Wrong. Customers call and ask, "Do you know that little plastic part that is shapped like a cone and has a tip going down at the end?", or "the thing that spins the disk?" It quickly became necessary to have a drive with the cover off sitting close by. Even with that the buyer wasn't always sure that was the part he needed. I guess one would think I was stupid for not knowing immediately the part needed. Please keep in mind, you're talking to a novice, therefore, I need all the help I can get.

Buzz's 142 commandment is; "if you have a piece of machinery, you should have a book on how it works." I'm inclined to agree with him. Besides, how are you going to run the alignment or diagnostic disk if you don't have a manual?

There use to be several manuals to choose from, but now the only available one is the 1084-02. It tells you how to run the diagnostic and alignment disk, as well as how to perform all sorts of tests and adjustments, has pictures of the drives showing the parts, schematics of the drive and the 1071 controller board, a parts list, and a discussion of the theory of operation for the whole system. This manual covers most MOD I and II drives and subsystems, and I assume will cover MOD IVs, even though they aren't specifically mentioned. DAMAN sells this two-inch thick manual for $47.50, plus shipping.

The second manual is a 1015. It is more or less the same as the 1084 without all the goodies I just mentioned. This manual is for those who want to send their drives off to have them repaired.

There is also a manual for the new 1115 drives. It's about an inch thick, has similar information to the 1084-02, except controllers and subsystems aren't discussed. DAMAN sells the 1115 manual for $23.75, plus shipping.

While I realize some of you don't want to do your own major maintenance, having the manual gives you that option as well as a source of information about a tool that performs an important function for you. Not having a manual is like trying to make a cake without a recipe. We'll try something but we'll not be sure of the results.

If any of you need parts, I'll be happy to help - but be prepared; I'll ask if you know the part number!

Later....

.....

CP/M CBIOS FILE

by William J. Leary Jr.
280 Ash Street, Hopkinton, Massachusetts 01748

After much ado and many postponements I have finally gotten my Customized BAsic Input Output System (CBIOS) for CP/M into order. I'd be delighted to see this released to the MUG Library for general distribution. Code for three programs, CBIOS, DISK, and SWAP, reside on this disk.

Requirements for installation are:

Microsoft M80 Macro Assembler
Microsoft L80 Linker
Zilog Z80 (or equivilant) processor
An INSTALLED CP/M system

CBIOS

This is actually three programs in one code file.

The CBIOS proper, along with the cold boot sectors necessary to load the CP/M, CCP, BDOS, and CBIOS into memory. Also there is an installation program that places the CBIOS in memory when CBIOS is executed. It can optionally overwrite the CBIOS and boot sectors on disk.

There are a few things to watch out for when assembling this program. Some of them are contained in declarations near the top of the program.

item 1:
```
msize  equ  62  ;kilobyte size of the CP/M system
```

This is used to cause the assembly to generate code ready to execute at the proper location. It is equated to the kilobyte size of the CP/M system the CBIOS is to support.

item 2:
```
fudge  equ  0500h   ;fudge factor for CentaCP/M
```

The CP/M system allows only 512 bytes to implement the CBIOS. This is never enough for Micropolis systems. Thus, everyone who brings up CP/M on Micropolis does one of two things.

1) Warn the owner to sysgen his system enough below the real top of ram to allow space for the oversize CBIOS.
2) Modify the MOVCPM program to cause it to place the system such that it allows for the oversize CBIOS itself.

My system, from Centa Systems, used the second approach and moved the system down 500(hex) bytes. The equate "fudge" records this amount. It may have to be changed or set to zero for other systems.

item 3:
At least on my system, the resultant area allotted for the CBIOS is 512 bytes larger than there is room for on the disk. Therefore, the CBIOS is arranged such that only uninitialized storage is up in this area. Things like the stack and the buffers. I've set up the install program such that it SHOULD abort the overwrite of the CBIOS if this size is exceded. However, I've not tested this trap and can't attest that it will work.

item 4:
My system runs at 4Mhz. I've set up the code such that it SHOULD run ok at 2Mhz, but I haven't tested this either. It was such a pain to upgrade my system to 4Mhz that I'm loth to set it back down to 2Mhz to test a feature that's useless to me.

item 5:
This CBIOS allows for the disk motors to be shut off when the drives are not in use. I know for a fact that this works because I've modified my disk controller board to shut off my motors after thrity seconds of non-use.

item 6:
This CBIOS has a drive mapping table that allows requests made via the normal jump vectors to be redirected to other drives. I've done this because I have 4 drives. Two in the computer, which is on a shelf kind of up out of reach, and two in an expansion cabinet right down on my desk. Swapping the A: and B: drives (in the computer) with the C: and D: drives (on the desk) makes it much more convienient for me to use my system. Besides that, CP/M has this habit of beating the A: drive to death. This swap moves the wear down to my C: drive.

DISK

This program has the following command format:

```
A>DISK COPY [TO] drive [FROM drive]
A>DISK FORMAT [TO] drive
A>DISK NEW [TO] drive [FROM drive]
A>DISK SHOW
A>DISK SWAP drive WITH drive
A>DISK SYSGEN [TO] drive [FROM drive]
A>DISK <ENABLE DISABLE> <PWI RAW ERR>
```

Where drive refers to a PHYSICAL drive. This program pays no mind to the drive map table in the CBIOS.

Where anything enclosed in [ ] pairs is optional. Not entering a FROM drive causes the program to use A:.

Where anything enclosed in < > pairs means you MUST pick one item from the group, ie: DISK ENABLE PWI

Where PWI means Pre-Write ID check. The track and sector ID are verified before each write.

Where RAW means Read-After-Write. Each sector is read after it is written to ensure it got on the disk correctly.

Where ERR means Error-Report and Retrys. If an error occurs the system will normally make a number of retrys, then report the error if the retrys fail.

Where the SHOW option will display the current status of the above three flags and also list the current disk map assignments.

Where SWAP exchanges any two PHYSICAL drives. Thus the command DISK SWAP A: WITH B: will cause all future A: accesses to be directed to the B: drive. If you then issued the command DISK SWAP B: WITH C: you would end up with A: accesses going to the C: drive. Remember it swaps the PHYSICAL drives.

Where NEW does a format followed by a sysgen.

SWAP

This program is just a quick and dirty job which swaps the A: and B: drives with the C: and D: drives. It works like the commands:

```
DISK SWAP A: WITH C:
DISK SWAP B: WITH D:
```

.....

Editor Note: Bill's files are on Library Disk 1911. However, this is another one that's way over my head. I imagine I'll have a million questions once I start trying to figure it all out. One thing for sure, I'll need to learn Zilog mneumonics.

.....

****
* *
* BASIC/Z SPECIAL INTEREST GROUP *
* *
****

BASIC/Z CORNER #20

by Steven Guralnick
375 South Mayfair Avenue, Suite 205
Daly City, CA 94015

I sent out some letters last month to folks who are using BASIC/z and got some nice replies. Apparently I am on to the right approach.

I believe I have mentioned this before but if not, I will say again that I will furnish any column on an eight inch SSSD disk to anyone who sends me one. If there's a program listing I've done you would like to have, just let me know. Buzz can offload an eight inch to five inch Micropolis. I will send the file as an ASCII file so you can load in in more easily.

Jerry Lenz has written two programs I will reproduce here. The first is a disk testing routine and the second lets you change one of the data records, to force an error and make sure the program is working correctly. I get more impressed every day with the functions available in this BASIC. Just take a look at some of them in this program, keeping in mind how much code does not have to be written because of them.

```
1  REM THIS PROGRAM WILL WRITE RECORDS ON TWO
   SEPARATE DRIVES. THE DISKS
2  REM SHOULD THEN BE SWITCHED AND READ BACK.
3  REM THE PROGRAM IS WRITTEN STRUCTURALLY.
4  REM ALL GOTO AND GOSUB COMMANDS ARE ADDRESSED TO
   @LABELS INSTEAD OF
5  REM LINE NUMBERS.
10 REM
20 SIZES(5,4,130,250)
30 DIM A$(130),REC.NO.FOUND(20)
40 ON ERROR GOTO @ERRSTOP:REM 'DEBUG "L" ' WILL
   RETURN LINE # WHERE ERROR OCCURS
50 GOSUB @CREDITS
60 UPCASE: REM ELIMINATES NECESSITY OF TRAPPING FOR
   LOWER CASE INPUTS
70 RESET: REM BE SURE TO RESET DRIVES
90 STRING CHR$(254): REM CHANGES STRING DELIMITER
   TO NON-PRINTING
100 LABEL @MENU:!**********************************
110 CLS
120 PRINT TAB(2,10);"***** DRTST *****"
130 PRINT TAB(3,5);"Program for testing disk drive
   compatability"
140 PRINT TAB(5,5);"1 - Instructions."
150 PRINT TAB(5);"2 - To prepare disks."
160 PRINT TAB(5);"3 - To read and test disks."
170 PRINT TAB(5);"4 - To exit from program."
180 PRINT:PRINT TAB(5);"Enter your choice.";
190 CHOICE$=EDIT$(1):PRINT"":REM I ALWAYS USE
   PRINT"" AS A STOPPER BEHIND EDIT$( )
200 IF CHOICE$="1" THEN GOSUB @INSTRUCTIONS
210 IF CHOICE$="2" THEN GOTO @SET.DRIVES
220 IF CHOICE$="3" THEN GOTO @COMPARE
230 IF CHOICE$="4" THEN CLOSE:GOTO @EXIT
240 GOTO @MENU
250 LABEL @SET.DRIVES:!****************************
260 CLS
270 GOSUB @SET.DISKS
280 PRINT:PRINT TAB(10);"***** SET.DRIVES *****"
290 PRINT TAB(5);"For setting up file #1 on drive
   #";DR1$
300 PRINT TAB(5);"And file #2 on drive  #";DR2$
310 PRINT:PRINT TAB(5);"Remove any  disks in drives
   ";DR1$;" and ";DR2$;" then"
320 PRINT TAB(5);"replace with formatted disks
   marked A and B."
330 GOSUB @CONTINUE
340 IF QQ$="E" THEN GOTO @MENU
350 RESET: REM RESET IS VERY HANDY
360 CLS
370 !
380 ! ******************************************
390 REM THE FOLLOWING ROUTINE WILL CREATE A NEW
   FILE. IF THE FILE EXISTS, THEN IT WILL
400 ! ENCOUNTER AN ERROR CONDITION WHICH WILL CAUSE
   IT TO SCRATCH THE FILE THEN
410 ! GO BACK TO CREATE A NEW FILE. THEN IT WILL
   JUMP OUT OF THE ROUTINE.
420 !
430 LABEL @CREATE1:CREATE 1 DR1$+"AFIL" RECLEN 128
   ERROR @SCRATCH1:GOTO @JUMP1
440 LABEL @SCRATCH1:SCRATCH DR1$+"AFIL":GOTO
   @CREATE1
450 LABEL @JUMP1
460 LABEL @CREATE2:CREATE 2 DR2$+"AFIL" RECLEN 128
   ERROR @SCRATCH2:GOTO @CLEAR.ERROR
470 LABEL @SCRATCH2:SCRATCH DR2$+"AFIL":GOTO @JUMP1
480 !
490 LABEL @CLEAR.ERROR:!***************************
500 REM YOU MUST CANCEL THESE AS YOU ARE FINISHED
   WITH THEM ELSE A LATER
510 !   ERROR CONDITION WILL CREATE A BOUNCE-BACK.
520 ON ERROR 1 CLEAR
530 ON ERROR 2 CLEAR
540 !
550 ! ******************************************
560 !
570 MOST#=MIN(FREE(1),FREE(2)):REM RETURNS MINIMUM
   FREESPACE BETWEEN DISKS
580 CLS
590 PRINT TAB(5,5);"How many records do you want to
   use for comparing. "
600 PRINT TAB(5);"You may enter up to
   ";INT(MOST#*1000/128);" records."
610 PRINT:PRINT TAB(5);"Enter amount ";
620 TOTAL#=VAL(EDIT$(5)):PRINT""
630 GOSUB @CONTINUE
640 IF QQ$="E" THEN @MENU
650 CLS
660 PRINT:PRINT:PRINT TAB(5);"You may stop the
   process by pressing <esc> key."
670 PRINT:PRINT TAB(5);"The counter will stop at
   ";TOTAL#
680 REM A$ HAS NO MEANING, IT IS ONLY A BUNCH OF
   GARBAGE
690 A$="ABCDEFGHIJKLMNOPQRSTUVWXY
   Z0123456789abcdefghijklmnopqrstuvwxyz012345678
   9ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghi
   jk"
700 REC$=FMT(I%,"99999"): REM REC$ HOLDS THE RECORD
   # TO 5 PLACES
710 FOR I%=1 TO TOTAL#
720 PRINT TAB(15,15);I%
730 REM FILES 1 & 2 ARE IDENTICAL
740 PUT 1 RECORD I% REC$+A$
750 PUT 2 RECORD I% REC$+A$
760 IF INCHAR$=CHR$(27) THEN GOSUB @CONTINUE:IF
   QQ$="E" THEN @MENU
770 NEXT I%
780 CLOSE
790 PRINT:PRINT TAB(5);"Task completed. Disks are
   ready for comparison. You may wish to"
800 PRINT TAB(5);"put these disks aside for later
   comparing."
810 GOSUB @CONTINUE
820 GOTO @MENU
830 LABEL @CONTINUE:!******************************
840 PRINT:PRINT TAB(5);"Press <cr> to continue,
   else E to exit ";
850 QQ$="":QQ$=EDIT$(1):PRINT""
860 RETURN
870 LABEL @INSTRUCTIONS:!**************************
880 CLS
890 PRINT TAB(3,10);"********** INSTRUCTIONS *****
   *****"
900 PRINT:PRINT TAB(5);"This program will write
   records to two separate disks"
910 PRINT TAB(5);"You may then switch the disks
   into different drives for checking"
920 PRINT TAB(5);"If the drives are not able to
   read properly then the alignment is suspect."
930 PRINT TAB(5);"As the testing takes quite a bit
   of time, you should set-up two formatted"
940 PRINT TAB(5);"disks marked with the drive names
   used when disks were written."
950 PRINT TAB(5);"You can then use these disks for
   later checking."
960 GOSUB @CONTINUE
970 RETURN
980 LABEL @SET.DISKS:!*****************************
990 CLS
1000 PRINT TAB(5,5);"Enter the drive name of one of
     the drives to be tested. ";
1010 DR1$=EDIT$(1):PRINT""
1020 PRINT:PRINT TAB(5);"Enter the drive name of
     the other drive to be tested. ";
1030 DR2$=EDIT$(1):PRINT""
1040 DR1$=DR1$+":":DR2$=DR2$+":": REM ADD COLON
1050 RETURN
1060 !
1070 LABEL @ERRSTOP:REM ***************************
1080 BELL(1)
1090 PRINT TAB(CLINE,12);"***** A  GLOBAL ERROR HAS
     TAKEN PLACE *****"
1100 PRINT TAB(CLINE,12);"Files will be closed and
     you are being returned":PRINT TAB(12);"to your
     operating system."
1110 PRINT TAB(CLINE,12);"The type of error is
     ";ERR$
1120 IF ERRLINE<>0 THEN PRINT TAB(CLINE,12);"Error
     occurred at line";ERRLINE
1130 IF LASTFILE<>0 THEN PRINT TAB(CLINE,12);"The
     last file accessed was ";LASTFILE;:IF
     STATUS(LASTFILE)=1 THEN PRINT "named
```

```
    ";NAME(LASTFILE) ELSE PRINT " which was
    closed."
1140 IF NAME$<>"" THEN PRINT
    TAB(CLINE,12);"[Drive:] Program which failed
    was [";SELECT$;"] ";NAME$: REM THE VALUE FOR
    NAME$ MUST BE IN EACH PROGRAM AHEAD OF THIS
    ROUTINE.
1150 PRINT TAB(CLINE,12);"Free memory left is
    ";SPACELEFT
1160 PRINT TAB(CLINE,12);"Free space left on disk
    ";SELECT$;"  ";FREE(SELECT$)*1000;" bytes"
1170 PRINT TAB(CLINE,12);"Address of error is
    ";ERRADDR
1180 PRINT TAB(14,12);"Files are now being
    closed.":CLOSE:STOP
1190 !
1200 LABEL @COMPARE:!*******************************
1210 CLS
1220 PRINT TAB(4,10);"***** COMPARE *****"
1230 ERROR.FOUND%=0
1240 GOSUB @SET.DISKS
1250 PRINT TAB(5);"Put disks to be read in drives
    ";DR1$;" and ";DR2$;"."
1260 PRINT:PRINT TAB(5);"The reading may be paused,
    or stopped by pressing <esc>."
1270 GOSUB @CONTINUE
1280 IF QQ$="E" THEN @MENU
1290 CLS
1300 RESET
1302 OPEN 1 DR1$+"AFIL" RECLEN 128
1304 OPEN 2 DR2$+"AFIL" RECLEN 128
1310 PRINT TAB(5);"Each record has a record length
    of 128  and files #1 & #2 will be compared"
1320 PRINT TAB(5);"byte for byte."
1330 PRINT TAB(5);"If an error is found, the
    program will pause. You may look at the
    defective"
1360 PRINT TAB(5);"record then continue or exit.
    Count will stop at ";SIZE(1)
1370 IF QQ$="E" THEN @MENU
1380 IF INCHAR$=CHR$(27) THEN GOSUB @CONTINUE:IF
    QQ$="E" THEN @MENU
1390 I#=1
1400  DO
1410    PRINT TAB(12,15);I#
1420     GET 1 RECORD I# A$
1430     GET 2 RECORD I# B$
1440    IF B$<>A$ THEN GOSUB @NOMATCH
1450   INCR I#
1460 UNTIL I#=SIZE(1)
1470 CLS
1480 PRINT:PRINT TAB(5);"** CHECK-OUT COMPLETED **"
1490 FOR I&=1 TO ERROR.FOUND%
1500 IF ERROR.FOUND%<>0 THEN PRINT:PRINT
    TAB(5);"RECORD # ";REC.NO.FOUND(I&);" DOES NOT
    MATCH"
1510 NEXT I&
1520 GOSUB @CONTINUE
1530 CLOSE:GOTO @MENU
1540 LABEL @NOMATCH:! ***************************
1542 REM IF AN ERROR WAS FOUND, THEN THIS ROUTINE
    WILL NOTE THE RECORD # OF THE ERROR
1550 CLS
1560 BELL(1):PRINT:PRINT"***** RECORD #";I#;" IS IN
    ERROR *****"
1570 INCR ERROR.FOUND%:REC.NO.FOUND(ERROR.FOUND%)=I
    #
1580 PRINT
1590 PRINT REC$;A$:PRINT:PRINT B$
1600 !
1610 REM ****************************************
1620 REM THIS ROUTINE SEARCHES A$ & B$ FOR LOCATION
    OF MISMATCH
1630 I%=0
1640 LABEL @INCRI:INCR I%
1650 DIFFA$=MID$(A$,I%,1)
1660 DIFFB$=MID$(B$,I%,1)
1670 IF DIFFA$=DIFFB$ THEN GOTO @INCRI
1680 PRINT:PRINT TAB(5);"The first occurrence is at
    position    ";I%
1690 PRINT
1700 REM ****************************************
1710 !
1720 GOSUB @CONTINUE
1730 IF QQ$="E" THEN GOTO @MENU
1740 CLS
1750 RETURN
1760 LABEL @CREDITS:!*******************************
1770 CLS
1780 PRINT TAB(5,10);"*******************************
    **************"
1790 PRINT TAB(10);"*
        *"
1800 PRINT TAB(10);"*                DRTST
        *"
1810 PRINT TAB(10);"*
        *"
1820 PRINT TAB(10);"*          Copyright (cr) 1984
        *"
1830 PRINT TAB(10);"*              G. B. LENZ
        *"
1840 PRINT TAB(10);"*
        *"
1850 PRINT TAB(10);"*******************************
    ************"
1860 FOR I%=1 TO 500:NEXT I%
1870 RETURN
1880 !
1890 LABEL @EXIT:!**********************************
1900 CLS
1910 PRINT:PRINT:PRINT"Before exiting, you must
    have a Systems disk in A drive."
1920 GOSUB @CONTINUE
1930 STOP
```

.....

```
10 REM ********** CHANGIT **********
20 REM A PROGRAM FOR CHANGING DATA IN "AFIL"
30 REM
40 SIZES(8,4,250,250)
50 STRING CHR$(254): REM CHANGE DELIMITER
60 @START
70 UPCASE
80 REM IT CAN BE ADAPTED TO WORK ON ANY RANDOM FILE
   BY CHANGING "AFIL" TO SUIT.
90 CLS
100 PRINT:PRINT:PRINT TAB(10);"***** CHANGEIT
    *****"
110 PRINT:PRINT"What drive has the file, else E to
    exit  ";:DR$=EDIT$(1):PRINT""
120 IF DR$="E" THEN STOP
130 DR$=DR$+":"
140 OPEN 1 DR$+"AFIL" RECLEN 0
150 PRINT:PRINT:PRINT "Enter record number to
    change, else E to exit. ";
160 REC$=EDIT$(5):PRINT""
170 IF REC$="E" THEN STOP
180 REC#=VAL(REC$)
190 LOWCASE
200 GET 1 RECORD REC# A$
210 PRINT:PRINT
220 A$=EDIT$(A$,128):PRINT""
230 PUT 1 RECORD REC# A$
240 CLOSE 1
250 GOTO @START
```

.....

Next month, I am going to tell you how you can get into a .COM or a .BZO file by using a very simple program written in BASIC/z and change something. I have used it to create a serial number inside TICKLER and it's very impressive. No external data file--just jam it right into the file. Probably completely unique to this BASIC.

See you soon.

(End - Basic/z SIG)

.....

****

CLASSIFIED
==========

WANTED: I have a VG 5005-1 and I am in the marked for a good used VG back-up that is compatible. Will consider a VG 3/VIP 2-drive system, but prefer a 5-megabyte system, without printer.

Dr. Robert L. Ward, Freedom Village Professional Center, 2815 Crisman Street, Suite A, Charlotte, NC 28208 (704) 399-0477.

.....

WANTED: Two TEI RAM 16 boards.

Jon Bopp, Mechanical Suppliers, Inc., 301 South 1st St., Terre Haute IN 47808 (812) 232-3333.

.....

FOR SALE: 1053 MOD II dual drive system, complete with controller and software. $500 postage paid in US.

William H. Ivey, 617 Langholm Dr., Winter Park FL 32789 (305) 677-0614.

.....

FOR SALE: Vector Graphic System B, two floppy drives, $2000. Vector 5032 32-megabyte system, $7000. (New) Vector-4; six months old, for sale at dealer cost. Includes Memorite and Execuplan.

Beverly Buckley (203) 847-0237.

.....

****
* DAMAN ADVERTISEMENTS *

GET YOUR FINANCES ORGANIZED

MINI-LEDGER
==========

for small business or personal finance

MINI-LEDGER is a single-entry accounting program developed for very small businesses (less than 25 employees). Typically, owners of such businesses do not need, do not want, and cannot justify the time needed to learn the larger, integrated computer accounting packages.

MINI-LEDGER was created to fill the gap between full accounting systems needed by larger companies and the often simplistic personal finance programs.

Specifically, the program assumes the business writes its checks by hand and does not want the computer to print them. At the same time, the program assumes the business wants to provide its accountant with a clear picture of how its money was spent so the expense of preparing taxes and other financial reports will be minimized.

While maintaining the usual Paradigm simplicity of menu-driven operation, MINI-LEDGER provides a powerful means of expense and income tracking by allowing 99 account and sub-categories. Several printouts (reports) are included which provide both the business owner and the accountant a clear picture of the company's financial operation.

MINI-LEDGER is written by Paradigm Consultants, requires CP/M, at least one disk drive, and a 132 column printer. It lists for $150. DAMAN sells it for $133, or $126.35 with your MUG and cash discounts.

.....

MAKE THE MOST OF YOUR COMPUTER

HYPERTYPER
==========

Becoming computer literate is increasingly essential these days. There are operating systems to learn as well as software packages. Home computer systems include home accounting software, word processing software, and a few games for the kids. But what if you have never learned to type? The answer is HyperTyper. HyperTyper can teach the fundamentals of typing or transfer skills from the typewriter or 10-key pad to the computer terminal. First it teaches the home keys and progresses to drills on new keys. Soon there are practice paragraphs. HyperTyper displays the speed and number of errors for the session.

You made 0 errors, with 100% accuracy, and
8 Words Per Minute.
Press <RETURN> to repeat, <ESC> for the menu:

The main menu of HyperTyper looks like this:

1)...Learn New Keys
2)...Practice Paragraph
3)...General Practice
4)...Numeric Keypad Practice
5)...End of Line Key: SPACE)
6)...Quit HyperTyper

There are exercises for operators who already know all the keys. HyperTyper even teaches the numeric ten key pad. The operator can practice in either "computer mode" or "typewriter mode". The "computer mode" simulates the wrap-around feature of a word processor, while the "typewriter mode" requires a carriage return at the end of each line.

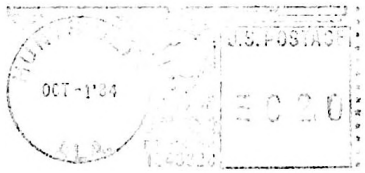
HyperTyper is available for any CP/M system, the IBM PC, or the Apple IIe with an 80-column card. It lists for $50. DAMAN sells it for $44.80, or $42.56 with your MUG and cash discounts.

.....

PURCHASING NOTES: Add $3 shipping per order.

==========

MICROPOLIS/VECTOR GRAPHIC
USERS GROUP
A Division of DAMAN
604 Springwood Circle
Huntsville AL 35803
(205) 881-1697



MR DAVE LAND 8507
5809 W 101ST STREET
OVERLAND PK KS 66207

FIRST CLASS MAIL
==========

FIRST CLASS MAIL
==========